

텍스트 분석 기반 대화 이력 추적을 이용한 작가 지원 기법

김 현 식*, 박 승 보**, 이 오 준***, 백 영 태****, 유 은 순**

Authoring Support Technique Using Text Analysis-based Dialogue History Tracking

Hyun-Sik Kim*, Seung-Bo Park**, O-Joun Lee***, Yeong-Tae Baek****, Eun-Soon You**

요 약

본 논문에서는 스토리의 논리적 오류를 막기 위한 방법으로 캐릭터의 대화 이력 구축과 추적 방법을 제안한다. 장편 소설이나 공동 집필 작품 등 작품의 길이가 길고 다수의 캐릭터가 등장하는 작품의 경우 작가에게 인지적 부담을 주게 되며 만약 작가가 캐릭터의 기억을 혼동할 경우 작품 속에 논리적 오류가 발생해 작품의 완전성과 무결성에 악영향을 미치게 된다. 본 논문에서는 상기한 추적 방법을 통해 작가의 인지적 부담을 경감하여 작품의 집필을 보조할 수 있는 작가 지원 시스템을 설계하고 기존에 집필된 소설을 분석하여 캐릭터의 대화 이력 추출 및 추적 예시를 보인다. 또한 질의문(query)의 성능평가를 통해 대화 이력 시스템이 평균 68.5%의 정확도를 보여주는 것으로 나타났다.

▶ Keywords : 저작 도구, 스토리, 캐릭터, 대화

Abstract

This paper suggests methods to chronicle and track the history of dialogues exchanged among characters to prevent logical errors of a story. As for stories that are long with many characters, especially in full-length novels and co-written stories, cognitive burden is imposed on a writer. If the writer has confused understanding of a character, then a logical error would enter the story. This would compromise completeness and integrity of writing. Against the backdrop, this paper

•제1저자 : 김현식 •교신저자 : 유은순

•투고일 : 2014. 8. 21, 심사일 : 2014. 8. 28, 게재확정일 : 2014. 9. 15

* Dankook University

** Dankook University Institute of Media Content, Dankook University

*** Dankook University Dept. of Software science, Dankook University

**** Kimpo University Dept. of Multimedia, Kimpo University

※ 이 논문은 2014년 한국컴퓨터정보학회 제50차 하계학술대회에서 발표한 논문("스토리의 플롯 구성 오류 추적을 위한 텍스트 분석 방법[12]")을 확장한 것임

shows how dialogues among characters are chronicled and tracked by using the aforementioned tracking methods through design of a writer support system that relieves a writer's cognitive burden while supporting the writing and through an analysis of existing novels. In addition, we showed the accuracy results of average 68.5% through the performance evaluation of the query used in the dialogue history tracking.

▶ Keywords : authoring tool, story, character, dialogue

I. 서 론

작가가 작품을 집필할 때 반드시 고려해야 하는 것이 스토리의 완전성과 무결성이다. 스토리의 완전성이란 스토리를 구성하는 모든 사건들이 작품이 끝날 때 충분히 매듭지어져야 한다는 것을 의미한다[1]. 그리고 스토리의 무결성이란 스토리의 사건들이 서로 모순 없이 시간에 따라 혹은 인과관계에 의해 연결되어야 함을 뜻한다[1,2]. 특히 작품이 연작(series)의 형태를 띠거나 많은 수의 캐릭터들이 등장하는 대화 장편 소설일 경우, 그리고 여러 명의 작가가 공동으로 집필하는 소설일 경우에 작가는 창작 과정에서 생산되는 수많은 정보들에 대해 혼란을 일으킬 가능성이 커지게 된다.

작품 속 캐릭터들은 어떤 사건이나 인물에 대해 동일한 정보를 갖고 있지 않다. 만일 작가가 각각의 캐릭터가 알고 있는 정보와 모르는 정보를 정확하게 인지하지 못하면 스토리의 논리적 오류를 범하게 되고 결국 스토리의 완전성과 무결성을 해칠 수 있다. 특히 독자에게 전달되는 정보를 의도적으로 제한하는 추리소설이나 스릴러 장르의 경우 극적 긴장감을 유지하고 독자의 몰입을 유도하기 위해서는 작가가 독자에게 직접 제공할 정보와 복선을 통해 암시해야 할 정보를 잘 조절해야 한다. 만일 이것이 실패하면 독자는 과소 정보로 스토리의 흐름을 이해하는데 어려움을 겪게 되거나, 과대 정보로 인해 긴장 상태를 유지할 수 없게 된다.

본 논문은 스토리의 논리적 오류를 탐지하고 작가들의 창작을 지원하기 위해 캐릭터들의 정보를 효과적으로 관리할 수 있는 캐릭터 대화 이력 추적 방법을 소개한다. 캐릭터 대화 이력이란 작품 속에서 캐릭터가 대화(dialogue)를 통해 말한 것과 상대방으로부터 들은 것을 뜻한다. 작가는 누가 누구와 어떤 내용의 대화를 했는지를 추적하여 캐릭터가 알고 있는

정보와 모르는 정보를 관리함으로써 스토리의 오류를 사전에 방지할 수 있다. 본 연구에서는 소설 텍스트를 이용하여 대화 문 분석을 통해 캐릭터의 대화 이력을 구축하고 추적하는 방법을 제안한다.

본 논문의 구성은 다음과 같다. II장에서는 저작도구와 캐릭터 이력 추적과 관련된 연구에 대해 설명하고, III장에서는 제안하는 캐릭터 대화 이력 구축과 추적 방법을 소개하며, IV장에서는 특정 소설을 이용한 실험 결과를 제시하고 그 결과를 평가한다. 마지막으로 V장에서는 결론 및 향후 연구를 기술한다.

II. 관련 연구

현재 작가의 창작을 지원하는 소프트웨어들이 국내외적으로 활발하게 개발되고 있다. 창작 과정은 크게 프리 프로덕션(Pre-Production), 프로덕션(Production), 포스트 프로덕션(Post-Production)으로 구분되는데, 기존의 국내외 저작 도구는 주로 앞의 두 단계인 프리 프로덕션과 프로덕션 단계를 지원하며, 포스트 프로덕션을 지원하는 연구에 대해서는 거의 이루어지지 않았다. 본 연구에서 제안하는 대화 이력 추적은 포스트 프로덕션 단계에서 활용하는 것을 목적으로 한다.

저작 도구들은 주로 스토리의 아이디어를 지원하거나, 스토리에 필요한 정보를 저장, 관리하는 툴로 사용된다. 또한 상업적으로 성공한 영화들의 시나리오를 분석하여 스토리 템플릿이나 가이드라인을 제공하기도 한다[3].

국외의 경우 라이트 브라더스(Write Brothers)사의 드라마티카 프로(Dramatica Pro)를 비롯하여 영화 스크립트 제작에 사용되는 파이널 드래프트(Final Draft), 스토리 뷰(Story View) 등이 있는데, 특히 높은 판매고를 기록하고 있는 드라마티카 프로는 창작 과정에서 작가가 만들어낸 이야기의 요소들을

하나의 스토리로 조직화 해주고, 질문과 응답 기능을 통해 스토리의 방향을 가이드 한다(4). 하지만 이러한 저작도구들은 작가의 집필 습관에 맞게 문서 형식과 스토리 작성을 돕는 관점에서 제작된 저작도구이다. 따라서 기존의 저작도구들은 특정 작업 이론을 따르고 있기 때문에 작가들의 자유로운 집필 방식을 제한하는 문제가 있다. 파이널 드래프트는 다양한 유형의 시나리오 견본 템플릿을 제공하며 글의 흐름을 검토하는 편리한 인터페이스를 보유하고 있다(5). 스토리 뷰는 장르에 따른 템플릿과 이야기 맵을 제공하는 것이 특징이다(6).

국내에서 연구된 초기의 저작도구 연구는 안경진의 프리프로덕션에 대한 연구를 들 수 있다(7). 안경진은 작가의 인지적 부담을 덜어주기 위해 프리 프로덕션 단계를 지원하는 시스템을 소개하였다. 이 시스템은 작가가 이야기 속의 중요한 사건과 캐릭터의 기본 정보 등을 프리 프로덕션에서 미리 설정하도록 하고 있다(11).

국내에서 이루어진 저작도구에 대한 최근의 연구는 류철균의 스토리 헬퍼 연구이다(3). 스토리 헬퍼는 흥행에 성공한 영화와 애니메이션의 스토리 분석을 통해 모티프를 추출하여 이를 데이터베이스화함으로써 작가의 시나리오 창작을 보조하는 소프트웨어이다(3). 스토리에 대한 독자의 인지적 과정을 고려한 연구도 있다. 권호창 등은 작품에 대한 독자의 능동적 해석을 고려하여 독자 모델을 활용한 새로운 창작 지원 연구를 제안하였다(8). 김현식 등은 영화의 스크립트를 장면(scene)별로 나누어 각 장면에 등장하는 캐릭터의 대화내용, 대화 빈도 등을 분석하는 캐릭터 이력 추적 시스템을 제안하였다(9). 이 시스템은 기존의 연구와는 달리 포스트 프로덕션 단계를 지원한다. 완성된 작품을 대상으로 하며, 역으로 스토리의 플롯과 캐릭터의 정보를 추적하기 때문에 역공학(Reverse engineering)적 성격을 띠고 있다.

최근에는 스토리의 비선형성(non linear)을 특징으로 하는 인터랙티브 스토리텔링 저작도구도 등장하고 있다(10,11).

기존의 저작 도구는 주로 시나리오 창작에 한정되어 있기 때문에 소설이나 수필과 같이 창작의 지원 영역을 다양한 장르로 확장할 필요가 있다.

III. 캐릭터 대화 이력 구축 방법

1. 장면(scene) 분리와 스토리 시간 순 장면 정렬

소설 텍스트의 구조는 보편적으로 그림 1의 형식을 따른

다. 소설에서 대화문은 따옴표로 표시한다. 대화문의 화자는 대화문과 같은 줄에 있거나 또는 대화문에 인접한 줄에 위치하며, 문법적으로 주어가 된다.

“땅은 좀 관측했나?” 토그람이 물었다.
 “관측경에 비치는 상의 최상단을 바라보게. 극지의 얼음 층 아래쪽 말일세.” 란스스쿠가 말했다. “물은 보통 저런 갈색이나 녹색을 띄지는 않지. 이 세계에서 다른 행성이 필요치 않다면 지금 바로 저길 향해야 할걸세.” 그들은 올그렌이 돌아올 때까지 저 먼 행성을 조사하기 시작해 그 특징을 스케치하고 있었다.
 “어떤가?”
 토그람은 이미 조타수 도제가 귀를 쾌활히 발딱 세운걸 알아차렸지만 일단 물어보았다.
 “저희 것을 제외하고는 이 세계 전체에서 하이퍼 드라이브 흔적은 없었고 하십시오!”
 올그렌이 씩 웃었다.
 (중략)그는 양손을 비볐다. 행성으로 강하할 순간을 기다리려니 애가 탔다.
 *
 “운동시간이야 벱.”
 아트 스나이더가 그를 불렀다. 다섯 승무원 중에서 그가 가장 참견쟁이였다.
 “알았수 판초 나오리.”
 헤르조그가 한숨을 내쉬었다. 그는 몸을 밀어 운동용 자전거로 향한 뒤 열심히 밟아댔다. 처음에는 느릿느릿했지만 점차 강도를 더 세게 했다. 이렇게 하면 자유 유행으로 뻘에서 칼슘이 줄어드는 걸 막을 수 있었다. 게다가 이건 고정 일과중 하나였다.
 멜리사 오프는 지구에서 오는 뉴스를 듣고 있었다. “페르난도 발렌주엘라가 어젯밤에 죽었데.” 그녀가 말했다.
 “그게 누구야?” 스나이더는 아무뎨은 아니었다. 하지만 헤르조그는 그랬고, 날 때부터 캘리포니아 팬이었다.
 “나 그 사람 옛 선수들 게임에서 한 번 본 적 있어. 우리 아버지랑 할아버지는 항상 그 사람 이야기를 하셨지.” 그가 말했다. “그 사람 니이가 얼마나 됐어 멜?”
 “79세.” 멜이 답했다.
 “그는 항상 몸무게가 너무 나갔지.” 헤르조그가 슬픈 목소리로 말했다.

그림 1. 일반적인 소설의 예문 - 해리 터틀도브: 가지 않은 길
 Fig. 1. A Typical Example of the Novel - Harry Turtledove: The road not taken

소설에서 시공간이 변할 때 작가는 일반적으로 빈 줄이나 챕터 이름, 특수문자 등으로 시공간의 변화를 독자에게 알려준다. 본 논문에서는 동일 장소와 동일 시간 내에서 이루어지는 일련의 대사를 ‘장면(scene)’으로 정의한다. 소설 속 장면의 흐름은 크게 스토리 시간(Story time)과 담화 시간(Plot time)이라는 두 시간축에 따라 배열된다(8).

스토리 시간은 작가가 창조한 스토리 세계에서 연대기적으로 흐르는 시간으로 등장인물들은 이 시간 축을 따라 사건을 경험한다. 담화 시간은 작가에 의해서 재구성된 시간이다. 독자는 이 시간축을 따라 사건을 경험한다. 스토리 시간과 담화 시간이 일치하지 않는 예로는 회상 장면을 들 수 있다. 만약 2014년 봄에서 시작한 장면 A가 회상 장면 B로 넘어가면서

1997년 겨울로 돌아갔다가 다시 2014년 봄 C로 복귀하는 식으로 소설이 쓰여 있다면 독자는 A-B-C의 순서로 사건을 경험하지만 등장인물은 B-A-C의 순서로 사건을 경험하게 된다. 등장인물은 스토리 시간에 따라 사건을 경험하지만 소설 텍스트는 담화 시간에 따라 쓰여 있으므로 기계 분석을 위해 작가는 필요에 따라 장면을 직접 정렬해 주어야 한다. 장면 정렬 예시는 그림 2에 나온 것과 같으며 정렬된 장면을 표현하는 방식은 그림 4에서 챗터명 위에 있는 어노테이션을 참고한다. \scene seqNo:3 next:4에서 \scene은 다음 \scene어노테이션이 나올 때까지의 텍스트 영역을 하나의 장면으로 정의하는 역할을 한다. seqNo는 해당 scene의 id값이며 next는 스토리 시간 순으로 장면을 재정렬 했을 때 다음에 이어지는 scene의 seqNo를 뜻한다.

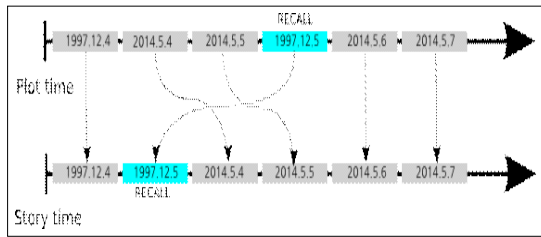


그림 2. 장면 정렬 예시
Fig. 2. Scene rearrange example

2. 화자-청자 대화 이력 작성

소설 텍스트를 장면으로 나누고, 스토리 시간으로 장면을 정렬한 후 시공간 안에 존재하는 모든 캐릭터를 자동으로 파악하여 화자-청자 대화 이력을 작성한다. 화자는 대사를 말한 캐릭터를 뜻하며 청자는 화자를 포함한 해당 장면에 존재하는 모든 캐릭터를 포함한다. 특정 장면에 등장한 캐릭터들의 집합은 해당 장면에 등장하는 모든 화자들의 집합으로 정의할 수 있으며, 이는 해당 장면의 모든 청자들의 집합과 같다.

$$C(S_i) = L(S_i) = \bigcup_{T_j \in S_i} T_j \quad (1)$$

S_i 가 i 번째 장면을, $C(S_i)$ 가 S_i 에 등장하는 캐릭터들의 집합을 의미할 때, $L(S_i)$ 는 S_i 에 등장하는 청자의 집합을, T_j 는 j 번째 화자를 의미한다. 하지만 한 마디도 하지 않고 오로지 청자 역할만을 수행한 캐릭터가 있을 수 있다. 이 경우에는 그림 4에서 처럼 장면 서두에 작가가 직접 등장인물 목록을 추가하는 어노테이션 작업을 해 주어야 한다.

2.1. 화자 판별

화자 판별은 대화문과 같은 줄 또는 바로 인접한 줄에 나타난 비 대화문의 주어를 감지하여 적용하며 대화문이 연속될 경우 두 사람간의 대화로 간주한다. 그림 3은 화자 판별을 위해 주어를 인식하는 과정을 나타낸 것으로 한국어의 경우, 주격 조사 '은', '는', '이', '가'를 통해 주어를 인식할 수 있다.

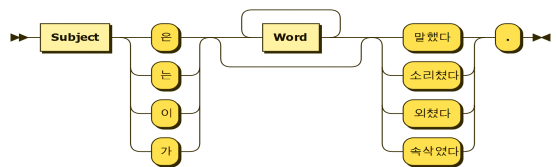


그림 3. 문장에서 주어를 탐색하는 방법(BNF)
Fig. 3. subject detection example

예를 들어 그림 4의 예문에서 "운동시간이야 벽."의 화자는 같은 줄에 나타난 비대화문의 주어인 '아트 스나이더'가 된다. 대화문의 처음 두 줄에 화자가 명시되면 그 뒤의 연속되는 대화는 이 두 사람의 대화로 자동 판단한다. 하지만 만일 화자가 명시되지 않은 경우에는 작가가 수동 어노테이션으로 화자를 명시한다.

그림 4에서 \character 어노테이션으로 표시한 헤르조그의 대화 "그 사람 나이가 얼마나 됐어 뽀?"의 경우 인접 서사문의 주어에 대명사 '그'가 쓰여 있어 현재의 분석 방법으로는 화자가 누구인지 알 수 없다. 그러나 작가는 이것이 헤르조그의 대화임을 알고 있으며 캐릭터의 대화 이력을 추적하기 위해 어노테이션을 붙인다.

표 1. 별명 테이블
Table 1. Nickname table

Real Name	Nickname(s)
아트 스나이더	스나이더
멜리사 오토	멜리사, 멜

소설 속 캐릭터들은 애칭 또는 예명을 갖고 있거나 화자 본인이 신분을 속이기 위해 가명을 쓰거나 이름을 바꾸는 경우가 있을 수 있다. 이 경우에 현재의 분석 알고리즘으로는 화자의 추정이 불가능하므로 이름을 연결하는 작업이 필요하다. 그래서 표1에서처럼 별명 테이블(Nickname table)로 처리한다. 별명 테이블은 동일인을 지칭하는 여러 이름을 리스트로 저장하는 역할을 한다.

```

Wscene seqNo:2, next:3
Wcharacters:토그람, 란시스크, 올그렌
(중략)
"땅은 좀 관측됐나?" 토그람이 물었다.
"관측경에 비치는 상의 최상단을 바라보게. 극지의 얼
음 층 아래쪽 말일세." 란시스크가 말했다. "물은 보통
저런 갈색이나 녹색을 띄지는 않지. 이 성계에서 다른
행성이 필요치 않다면 지금 바로 저길 향해야 할걸세."
그들은 올그렌이 돌아올 때까지 저 먼 행성을 조사하기
시작해 그 특징을 스케치하고 있었다.
"어떤가?"
토그람은 이미 조타수 도제가 귀를 꽤 팔짝 발짝 세운걸
알아차렸지만 일단 물어보았다.
Wcharacter:올그렌 "저희 것을 제외하고는 이 성계 전
체에서 하이퍼 드라이브 흔적은 없었다고 하십니다!"
올그렌이 씩 웃었다.
(중략)그는 양손을 비볐다. 행성으로 강하할 순간을 기
다리려니 애가 뒀다.
*
Wscene seqNo:3, next:4
Wcharacters:스나이더, 헤르조그, 멜리사,
"운동시간이야 벱."
아트 스나이더가 그를 불렀다. 다섯 승무원 중에서 그
가 가장 찡찡꿍이었다.
"알았수 판초 나으리."
헤르조그가 한숨을 내쉬었다. 그는 몸을 밀어 운동용
자전거로 향한 뒤 열심히 밟아댔다. 처음에는 느릿느릿
했지만 점차 강도를 더 세게 했다. 이렇게 하면 자유
유영으로 배에서 칼슘이 줄어드는 걸 막을 수 있었다.
게다가 이걸 고정 일과중 하나였다.
멜리사 오트는 지구에서 오는 뉴스를 듣고 있었다. "페
르난도 발렌주엘라가 어젯밤에 죽었다." 그녀가 말했다.
"그게 누구야?" 스나이더는 야구팬은 아니었다. 하지만
헤르조그는 그랬고, 날 때부터 캘리포니아 팬이었다.
Wcharacter:헤르조그 "나 그 사람 옛 선수를 게임에서
한 번 본 적 있어. 우리 아버지랑 할아버지는 항상 그
사람 이야기를 하했지." 그가 말했다. Wcharacter:헤르
조그 "그 사람 나이가 얼마나 됐어 땀?"
"79세." 멜이 답했다.
"그는 항상 몸무게가 너무 나갔지." 헤르조그가 슬픈
목소리로 말했다.
    
```

그림 4. 어노테이션 예시
Fig. 4. Annotation example

작품에 따라 같은 장면(시공간)에 존재했지만 화자의 대사를 듣지 못한 청자가 존재할 수는 있으나 이 시스템에서는 고려하지 않고 같은 시공간에 존재한 청자라면 무조건 해당 공간에서 발화된 모든 대사를 들었다고 가정한다. 소설은 시나리오와 달리 대사(dialog)가 있을 경우 화자가 누구인지 명시적으로 표시되어 있지 않다. 대사가 2개 이상 연속으로 나올 수도 있으며, 대사앞에 청자가 표현될 수도 있다. 따라서 소설에서 화자를 정확하게 추출하기 위한 명확한 방법이 필요하다. 그림 5는 연속되는 대화에서 화자 추적 알고리즘을 나타낸 것이다.

분석을 통해 도출된 화자-청자 이력 결과는 DB로 저장된다. 그림 6은 그림 1의 소설 예문의 화자-청자 이력 결과를 보여주고 있다. 화자는 자신이나 다른 사람이 말할 때 그걸 듣는 사람으로써, 동일한 시공간에 존재할 경우 모두가 청자가 될 수 있다. 따라서 청자는 동일 장면에서 발견된 모든 화

```

current_teller: String; //발견된 화자
teller: String; //현재 대사(dialog)의 화자
previous_teller: String; //이전 화자
prev_prev_teller: String; //이전의 이전 화자

Function Teller_detection(text):
for each line of text:
if line contains dialog then: //문장이 대사를 포함하면
find adjacent description: //주변의 문장을 검색
//현재 라인 중 1개를 검색
find one that present same line
//앞 뒤의 문장을 검색
find previous or next line(depending on writing
style) that contains description only
//주변의 문장이 있을 경우
if found adjacent description then
extract subject //주어를 추출
//주어가 현재 화자(current teller)가 됨
set current_teller to subject
//current_teller가 대사의 화자(teller)가 됨
set teller of this dialog to current_teller
if previous_teller has already been set
//이전 화자(previous_teller)는 이전의 이전 화자
//prev_prev_teller로 설정됨
set prev_prev_teller to previous_teller
//previous_teller는 current_teller로 변경됨
set previous_teller to current_teller
//current_teller가 검색되지 않았다면
if not found then
//이전 화자가 있다면
if previous_teller present then
//이전 화자가 현재 대사의 화자가 됨
set teller of this dialog to previous_teller
//이전의 이전 화자는 이전 화자와 서로 바뀜
swap prev_prev_teller and previous_teller
//이전 화자가 없다면 오류발생
if previous_teller not set then
report ambiguous error
    
```

그림 5. 화자 청자 추출 알고리즘
Fig. 5. Algorithm for detecting teller and listener

scene	lineNo	Teller	Listener	Dialogue
2	44	토그람	토그람, 란시스크, 올그렌	"그만 좀 칭칭거리게." 도
2	50	란시스크	토그람, 란시스크, 올그렌	네가 토그람이 하는 험담
2	51	토그람	토그람, 란시스크, 올그렌	맛대로 지껄어라지.
2	57	올그렌	토그람, 란시스크, 올그렌	"이 항성계에는 행성이 있
2	58	란시스크	토그람, 란시스크, 올그렌	어디 보지.
2	62	란시스크	토그람, 란시스크, 올그렌	"이거 행성이로구먼." 한
2	64	란시스크	토그람, 란시스크, 올그렌	"올그렌이 낙담하는 모습:
2	65	토그람	토그람, 란시스크, 올그렌	어디 한번 해보게나.
2	71	란시스크	토그람, 란시스크, 올그렌	암마, 대위는 너보다 더 많
2	75	올그렌	토그람, 란시스크, 올그렌	"녹색으로 덮여있는 행성!
2	79	란시스크	토그람, 란시스크, 올그렌	그렇지도 모르겠군.
2	83	란시스크	토그람, 란시스크, 올그렌	하지만 물이 많은 것처럼
2	84	토그람	토그람, 란시스크, 올그렌	거 루오프에는 아주 기
2	90	란시스크	토그람, 란시스크, 올그렌	"여기 있군." 그가 부드
2	91	올그렌	토그람, 란시스크, 올그렌	"아, 네, 넬!" 그의 도제
2	92	란시스크	토그람, 란시스크, 올그렌	슬레븐 전쟁관계 가서 어
2	96	란시스크	토그람, 란시스크, 올그렌	여기 와서 직접 보게.
2	100	토그람	토그람, 란시스크, 올그렌	"땅은 좀 관측됐나?" 토
2	101	란시스크	토그람, 란시스크, 올그렌	"**관측경에 비치는 상의
2	106	토그람	토그람, 란시스크, 올그렌	어떤가?
2	110	올그렌	토그람, 란시스크, 올그렌	저희 것을 제외하고는 이
3	125	스나이더	스나이더, 헤르조그, 멜리사	운동시간이야 벱.
3	129	헤르조그	스나이더, 헤르조그, 멜리사	알았수 판초 나으리.

그림 6. 화자-청자 분석 결과
Fig. 6. The Result for Analyzing Teller-Listeners

자가 된다.

장면 2의 각 문장들을 그림 5의 알고리즘과 위 방법에 의해 화자와 청자를 추출한 후에 정리하면 화자와 청자와 그들이 나누는 대사는 그림 6과 같이 나타나게 된다.

3. 캐릭터 대화 이력 추적

작가는 특정 캐릭터가 듣거나 말한 모든 대사를 화자-청자 이력 결과 DB를 통해 찾아낼 수 있다. DB에서 원하는 캐릭터의 이력을 추적하기 위해서는 화자와 청자와 대사의 데이터로 조합될 수 있는 다양한 경우에 맞는 질의문을 처리할 필요가 있다. 이를 위해 다음에 나오는 4개의 질의문을 설계하였다.

그림 6에서 Scene 번호가 같은 모든 셀의 Listener는 같은 값을 갖는다. 그림 6에서 보듯이 Scene 3의 Listener는 스나이다, 헤르조그, 멜리사 세 명이므로 Scene 2에 존재하는 토그람은 Scene 3에서의 대화를 모른다고 결정할 수 있다.

예를 들어 특정한 청자 A가 등장하는 장면의 집합은 청자 A가 등장한 장면들을 원소로 하는 집합으로 식 2와 같이 정의할 수 있다.

$$S(L_A) = \{S_i | S_i \ni L_A\} \tag{2}$$

L_A 는 A번째 청자를, $S(L_A)$ 는 L_A 가 등장한 장면의 집합을 의미한다.

그리고 특정한 청자 A와 B가 모두 등장하는 장면의 집합은 청자 A가 등장한 장면들의 집합과 청자 B가 등장한 장면들의 집합의 교집합으로 식 3과 같이 정의할 수 있다.

$$S(L_A | L_B) = S(L_A) \cap S(L_B) \tag{3}$$

만일 청자 A가 알고 있지만 청자 B는 모르는 정보를 조회하고 싶은 경우, 청자 A가 참여한 대화 집합과 청자 B가 참여한 대화 집합의 차집합으로 식 4와 같이 구할 수 있다.

$$K(\sim L_A | L_B) = D(L_A) - D(L_B) \tag{4}$$

L_A 가 A번째 청자를, $K(\sim L_A | L_B)$ 가 L_A 는 알지만 L_B 는 모르는 정보를 나타낼 때, $D(L_A)$ 는 L_A 가 들은 대화의 집합을 나타낸다.

특정 정보 a를 알고 있는 캐릭터들의 집합은 해당 정보를 포함하고 있는 대화들의 청자들의 합집합으로 식 5와 같이 표현할 수 있다.

$$C(K_a) = L(K_a) = \bigcup_{L_i \in D(K_a)} L_i \tag{5}$$

K_a 가 a번째 정보를, $C(K_a)$ 가 K_a 를 알고 있는 캐릭터들의 집합을 의미할 때, $L(K_a)$ 는 K_a 를 들은 청자들의 집합, $D(K_a)$ 는 K_a 가 포함된 대화를 나타낸다.

IV. 실험 및 평가

캐릭터 대화 이력 추적을 위해 구축된 실험 환경은 Ubuntu 계열 OS인 Linux mint 17 cinnamon에서 Maria DB 5.5.37버전을 사용했으며 질의문(query) 입력 및 결과 조회를 위해 phpmyadmin을 사용했다. 분석을 위한 원본 소설은 해리 터틀도브의 작품인 <가지 않은 길>의 한국어 번역판을 사용했다.

구현 여부 확인을 위하여 위에서 제시한 식(5)의 질의문을 테스트했다. 식(5)는 특정 정보 a를 알고 있는 캐릭터들을 추출하는 것으로 청자들의 합집합을 추출하는 것이다. 그림 7과 같이 특정 단어 “녹음기”를 들은 캐릭터 목록과 장면 결과를 검색하는 실험을 진행했다. 위 소설에서 “녹음기”라는 단어는 핵심 단어 즉 ‘키워드’이며 작품 속에서 큰 비중을 차지하면서 드물게 등장하는 단어에 해당한다.



그림. 7. 단어 “녹음기”를 들은 캐릭터 목록
Fig. 7. characters list hearing the word “recorder”

“녹음기”라는 단어를 들은 캐릭터는 장면 20에 등장하는 토그람과 힐다, 그리고 장면 22에 등장하는 토그람과 란시스크임을 알 수 있다.

```
SELECT distinct scene FROM `The_road_not_taken` a
WHERE (Listener like '%토그람%' or Listener like '%란시스크%') and a.scene not in (select distinct scene from `The_road_not_taken` where Teller = '올그렌')
```

```
SELECT DISTINCT scene
FROM `The_road_not_taken` a
WHERE (
  Listener LIKE '%토그람%'
  OR Listener LIKE '%란시스크%'
)
AND a.scene NOT
IN (
```

프로파일링 [인라인] [수정]

보기 : 시작 행: 0 행 갯수: 30 100 행마다 헤더 반복

키로 경찰: 없음

+ 옵션

scene

1

8

12

14

16

20

22

그림 8. 토그람과 란시스크는 알지만 올그렌은 모르는 정보에 대한 질의문과 질의 결과

Fig. 8 The Query and information that Togram and Ransisc know but Olgren doesn't know

그림8은 식(4)에 대한 실험 결과로 토그람과 란시스크는 알지만 올그렌은 모르는 정보에 대한 검색 결과이다.

그림 8에서 보듯이 토그람과 란시스크가 함께 등장한 장면은 1, 8, 12, 14, 16, 20, 22번이며, 올그렌은 이 장면 속에는 등장하지 않았기 때문에 올그렌은 토그람과 란시스크가 대화한 내용을 알지 못한다는 것을 검색 결과로 알 수 있다.

반대로 올그렌은 알지만 란시스크가 모르는 정보를 같은 방법으로 조회하면 한 건도 조회되지 않는다. 이를 통해 올그렌은 항상 란시스크와 함께 행동하는 캐릭터임을 유추할 수 있다.

개발된 쿼리의 성능평가를 위해 작가들이 창작 과정에서 필요로 할 것으로 예상되는 캐릭터 이력과 관련된 질문 중 비교적 복잡한 것들을 뽑아 표 2와 같이 작성하였다. 여기서 '관련된 단어'는 '동일한 의미를 갖는 단어'와 동의어로 사용했다.

표 2. 성능평가를 위한 질문들
Table 2. Questions for Evaluation

번호	질문	평가 방법
Q1	'캐릭터 C1, C2와 동시에 행동하는 제 3의 캐릭터	10명의 Character를 대상으로 45개의 질의문을 실행
Q2	'단어 W1'을 들었지만 뒤에 '단어 W1'과 관련된 단어를 한 번도 말하지 않는 캐릭터	10개의 단어에 대해 질의문을 실행
Q3	'캐릭터 C1의 행동 A1'을 보고도 행동과 관련된 단어를 한 번도 말하지 않는 캐릭터	10개의 단어에 대해 질의문을 실행

표 2와 같이 3개의 질문에 대한 질의문을 생성하여 총 65회의 질의 실행을 하였다. 정확도 100%의 기준은 사람이 수작업으로 직접 확인한 결과와의 차이의 절댓값으로 했다. 실험 결과 그림 9와 같이 Q1과 Q2에 대해서는 비교적 높은 결과를 얻었다. 오차는 대부분 대사문의 주어 탐지 알고리즘의 한계로 인해 특정 대사문의 화자를 추정하지 못했거나 잘못 추정한 경우에 발생했으며 그 외 해당 장면에서 한 마디도 하지 않은 캐릭터가 존재해 청자 목록에서 누락됨으로써 발생했다. Q3의 경우 설계된 데이터베이스가 대화를 기반으로 이루어졌기 때문에 따옴표 바깥에 존재하는 행동과 관련된 단어가 대사문에도 똑같이 들어 있는 경우가 아니면 검색되지 않아 매우 낮은 정확도 결과를 나타내었다.

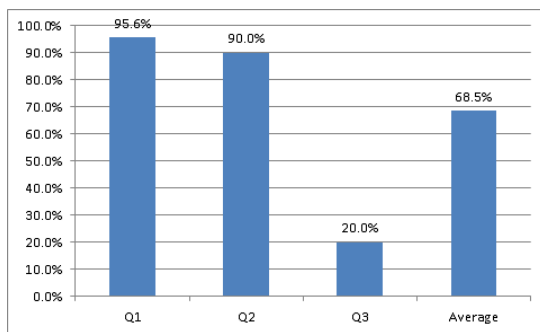


그림 9. 질문에 대한 성능평가 결과
Fig. 9 Result of evaluation for questions

평가에서 보듯이 캐릭터의 기억 이력을 완전히 추적하기 위해서는 캐릭터의 대화뿐만 아니라 그 행동 역시 추적할 필요가 있을 것이다.

위에서 지적했듯이, 기존의 저작 도구들은 창작의 세 단계 중 주로 프리 프로덕션과 프로덕션 단계에 집중되어 있으며, 작품에 대한 오류 수정이나 편집과 같은 포스트 프로덕션을 지원하는 도구의 개발이나 연구는 거의 이루어지지 않았다. 본 연구에서 소개한 대화 이력 추적 시스템은 포스트 프로덕션 단계에서 캐릭터들의 정보를 효율적으로 관리하여 작가의 인지적 부담을 줄여줄 수 있다는 점에서 기존의 연구와 차별성을 지닌다.

V. 결론

그동안 작가들은 본인의 기억에 의존하거나 작가 개인의 집필 노트 등을 통해 캐릭터 정보들을 관리해왔다. 본 논문에서는 이미 작성을 마친 원고를 분석하여 캐릭터의 대화 이력을 추적하고 요약해 보여줌으로써 작가의 인지적 부담을 덜어

주는 방법을 제안했다. 소설 <해리 터틀도브 : 가지 않은 길>에 대해 화자와 청자, 그리고 대화 내용을 추출하여 DB로 구축하고 설계한 4개의 질의문을 통해 작가가 창작 과정에서 필요한 정보를 알 수 있도록 구현하였다. 또한 3개의 비교적 복잡한 질문에 대해 총 65개의 질의문을 작성하여 평균 68.5%의 정확도 평가 결과를 얻었다. 대화에 대한 이력추적(Q1, Q2)에서는 92.8%의 정확도를 나타냈지만 행동기반의 이력추적 성능은 20.0%의 정확도가 나타났다. 이것은 알고리즘 자체가 대화만을 추적하고 있기 때문에 캐릭터의 행동에 대해서도 데이터 수집이 필요할 것으로 판단된다. 따라서 향후에는 대화와 행동 모두에 대해 이력 추적이 가능한 작가 지원 시스템에 대한 연구를 진행할 필요가 있다.

Acknowledge

이 논문은 2013년도 정부(교육부)의 재원으로 한국연구재단의 지원을 받아 수행된 기초연구사업임 (NRF-2013R1A1A2057943)

참고문헌

- [1] Aristotle, *Poetics*, Dover Publications, 1997.
- [2] R. C. Allen, "Television and Contemporary Criticism," The University of North Carolina Press, 1992.
- [3] C.-G. Lyou, and Y. Jeong, "A Study about <The Principle of Causality> of Digital Narrative Tool," Korea Humanities Content Society, Vol. 22, pp. 183-207, July 2011.
- [4] C.-G. Lyou, "A Study on the Narrative Algorithm of Story Creation Program : Focusing on <Dramatica Pro>," The Journal of Korean Fiction Research, Vol.-No. 38, pp. 117-152, May 2008
- [5] http://en.wikipedia.org/wiki/Final_Draft_software
- [6] <http://creative-writing-software-review.toptenreviews.com/story-view-review.html>
- [7] K.-J. Ahn, "Analyzing writer's storytelling task and designing a support system", College of Interdisciplinary Studies, Culture Technology Program, April 2006.
- [8] H. Kwon, H. Kwon, and W. Yoon, "Designing a Writing Support System Based on Narrative Comprehension of Readers," HCI 2014, pp. 405-408, 2014.
- [9] H.-S. Kim, E.-S. You, S.-B. Park, S.-J. Cha, and G.-S. Jo, "Character History Tracking System for Screenplay Writer," The Korea Multimedia Society, Vol. 13. No. 2, pp. 155-158, Nov. 2010.
- [10] E-R, Lee, K-C, Kim, "A Study on Implementation of Writing Supporting System(ICWS) for Interactive Storytelling Contents", Digital Policy Research, Vol.11, No.2, pp.263-269. Feb. 2013
- [11] E-R, Lee, K-C, Kim, "A Study on the Development of Intelligent Contents and Interactive Storytelling System", Digital Policy Research, Vol.11, No.1, pp.423-430, Jan. 2013
- [12] H.-S. Kim, S.-B. Park, Y. T. Baek, E.-S. You, "Fiction text analysis for plot error tracking", Proceedings of The Korea Society of Computer and Information Summer 2014, pp. 37-39, July 2014.

저 자 소 개



김 현 식
 2009: 인하대학교
 컴퓨터정보학과 공학사.
 2011: 인하대학교
 정보공학과 공학석사.
 현 재: 단국대학교 연구원
 관심분야: 컴퓨터공학
 Email : wbstory@storymate.net



박 승 보
 1995: 인하대학교
 전기공학과 공학사.
 1997: 인하대학교
 전기공학과 공학석사.
 2011: 인하대학교
 정보공학과 공학박사
 현 재: 단국대학교
 미디어콘텐츠연구원 연구원
 관심분야: 멀티미디어 정보검색,
 스토리텔링, 감정 검출
 Email : molaal@naver.com



이 오 준
 현 재: 단국대학교
 소프트웨어학과 학부 재학중
 전자계산학과 공학사.
 관심분야 : 추천시스템, 기계학습,
 정보검색, 적응형시스템
 Email : concerto_grs@naver.com



백 영 태
 1989: 인하대학교
 전자계산학과 이학사.
 1993: 인하대학교
 전자계산공학과 공학석사
 2002: 인하대학교
 전자계산공학과 공학박사
 1993-1998 : 대성정보기술(주)
 정보통신연구소
 선임연구원
 1998-현 재: 김포대학교
 멀티미디어과 부교수
 관심분야: 멀티미디어 정보검색,
 웹교육 시스템,
 모바일 시스템
 Email : hannaek@kimpo.ac.kr



유 은 순
 1995년: 인하대학교
 불어불문학과 학사
 2000년: 프랑스 브장송대학교
 언어학 석사
 2007년: 프랑스 브장송대학교
 언어학 박사
 2011년~2012년: 단국대학교
 미디어콘텐츠연구원
 전임연구원
 2012년~현재: 단국대학교
 미디어콘텐츠연구원
 리서치펠로우
 관심분야: 디지털 스토리텔링,
 온톨로지, 소셜미디어, 감정
 Email : tesniere@naver.com